Managing multi-reader medical imaging annotations with CVAT

Project Management and Software Development
for Medical Applications

General Info

Contact Person: Mickael Tardy, CSO, Hera-MI

Contact Email: Mickael.tardy@hera-mi.com

Project Abstract

Medical imaging annotation is more challenging as it often requires collecting the opinions of several readers for the same sample seeking to maximize the quality of the annotations and labels. In this work we will focus on CVAT, an open-source tool, to allow for an effective image annotation process.

Background and Motivation

Annotating medical images is often less obvious than natural ones. That is, the objects to be described (e.g., abnormalities, pathologies), are difficult to capture, and require clinical expertise, so we often need opinions from several experts for the same sample. This leads to a need of performing a multi-reader annotation process, with tracking and comparison of the collected annotations and labels.

CVAT (https://github.com/openvinotoolkit/cvat) is a popular open-source image annotation tool offering rich features for efficient image labeling. While working with single-track annotations is quite straightforward, the generic nature of CVAT does not allow to easily handle multi-reader annotations out of the box.

In this work, we would like to overcome the limits and lacks of CVAT to create an intuitive and effective toolset allowing to handle multi-reader annotations.

Student's Tasks Description

In this work, the student will be able to explore a truly full-stack nature of data management. Having deployed CVAT in Docker environment, we will first focus on the merging of annotations coming from different readers. Python code will need to be written to efficiently handle xml and json files.

Then, we will switch to activity tracking, where we will aim to better identify the authors of annotations. At this stage, the student will have a chance to look under the hood of an open-source project such as CVAT, exploring its Python backend and postgres database. The student will need to build a toolset on top of CVAT allowing for tracking user activity.

Technical Prerequisites

Full-stack software engineering:

- frontend: Javascript (required)

- Infrastructure: Docker (strongly desirable)

- backend: Python (required),

- database: Mongodb or Postgresql (desirable)

References

- Official CVAT github repository https://github.com/openvinotoolkit/cvat

Please send the completed proposal to ardit.ramadani@tum.de, tianyu.song@tum.de, vanessag.duque@tum.de and shervin.dehghani@tum.de. Please note that this proposal will be evaluated by the BMC coordinators and will be assigned to a student only in case of acceptance.